**Câu 1:** Biến int \*p là gì?  
A. Con trỏ trỏ tới kiểu char  
B. Con trỏ trỏ tới kiểu int  
C. Biến lưu giá trị kiểu int  
D. Không hợp lệ  
**Đáp án:** B

**Câu 2:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int a = 10, \*p = &a;

printf("%d", \*p);

A. 10  
B. Địa chỉ của a  
C. Lỗi biên dịch  
D. Không xác định  
**Đáp án:** A

**Câu 3:** Tên của một mảng trong C là:  
A. Giá trị đầu tiên của mảng  
B. Kích thước mảng  
C. Địa chỉ của phần tử đầu tiên  
D. Không xác định  
**Đáp án:** C

**Câu 4:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int arr[3] = {1, 2, 3};

int \*p = arr;

printf("%d", \*(p + 1));

A. 1  
B. 2  
C. 3  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 5:** Tại sao cần sử dụng con trỏ NULL?  
A. Để chỉ ra rằng con trỏ không trỏ tới đâu  
B. Để tăng hiệu suất chương trình  
C. Để tối ưu hóa bộ nhớ  
D. Để tránh lỗi logic  
**Đáp án:** A

**Câu 6:** Đoạn mã sau in ra gì?

|  |
| --- |
|  |

int x = 5;

int \*ptr = &x;

\*ptr = 10;

printf("%d", x);

A. 5  
B. 10  
C. Địa chỉ của x  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 7:** Phép toán nào cho phép lấy địa chỉ của một biến?  
A. \*  
B. &  
C. sizeof  
D. Không phép toán nào đúng  
**Đáp án:** B

**Câu 8:** Con trỏ không thể được sử dụng để:  
A. Truy cập địa chỉ của một biến  
B. Truy cập giá trị tại địa chỉ  
C. Thay đổi địa chỉ bộ nhớ vật lý  
D. Truy cập giá trị trong mảng  
**Đáp án:** C

**Câu 9:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int a = 5, b = 10;

int \*p = &a;

p = &b;

printf("%d", \*p);

A. 5  
B. 10  
C. Địa chỉ của b  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 10:** Đoạn mã sau có lỗi gì?

|  |
| --- |
|  |

int \*p;

printf("%d", \*p);

A. Lỗi biên dịch  
B. Không lỗi  
C. Truy cập vào con trỏ không được khởi tạo  
D. Lỗi runtime  
**Đáp án:** C

**Câu 26:** Đoạn mã sau có kết quả gì?

|  |
| --- |
|  |

int x = 10;

int \*p1 = &x, \*\*p2 = &p1;

printf("%d", \*\*p2);

A. 10  
B. Địa chỉ của x  
C. Địa chỉ của p1  
D. Lỗi biên dịch  
**Đáp án:** A

**Câu 27:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int arr[5] = {1, 2, 3, 4, 5};

int \*p = arr + 2;

printf("%d", \*p);

A. 1  
B. 2  
C. 3  
D. Lỗi runtime  
**Đáp án:** C

**Câu 28:** Con trỏ void \* có thể:  
A. Trỏ tới mọi kiểu dữ liệu  
B. Chỉ trỏ tới kiểu void  
C. Không sử dụng được  
D. Lưu giá trị kiểu int  
**Đáp án:** A

**Câu 29:** Đoạn mã sau in ra gì?

|  |
| --- |
|  |

int arr[3] = {1, 2, 3};

int \*p = arr;

printf("%p", p + 1);

A. Địa chỉ của phần tử thứ nhất  
B. Địa chỉ của phần tử thứ hai  
C. Giá trị của phần tử thứ hai  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 30:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int a = 5;

int \*p = &a;

int \*\*pp = &p;

printf("%d", \*(\*pp));

A. Địa chỉ của a  
B. 5  
C. Địa chỉ của p  
D. Lỗi runtime  
**Đáp án:** B

Câu 31: Đoạn mã sau có lỗi gì?

|  |
| --- |
|  |

int \*p = (int \*)malloc(sizeof(int));

\*p = 10;

free(p);

printf("%d", \*p);

A. Lỗi biên dịch  
B. Không lỗi, kết quả là 10  
C. Lỗi runtime do sử dụng con trỏ bị giải phóng  
D. Lỗi do không ép kiểu khi dùng malloc  
**Đáp án:** C

**Câu 32:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

char str[] = "Hello";

char \*p = str;

printf("%c", \*(p + 1));

A. H  
B. e  
C. l  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 33:** Địa chỉ mà một con trỏ NULL trỏ tới là:  
A. Địa chỉ hợp lệ trong bộ nhớ  
B. Địa chỉ 0  
C. Địa chỉ không xác định  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 34:** Đoạn mã sau in ra gì?

|  |
| --- |
|  |

int arr[5] = {10, 20, 30, 40, 50};

int \*p = arr;

p += 3;

printf("%d", \*p);

A. 10  
B. 30  
C. 40  
D. Lỗi biên dịch  
**Đáp án:** C

**Câu 35:** Khi nào cần sử dụng free trong lập trình con trỏ?  
A. Khi sử dụng malloc để cấp phát bộ nhớ  
B. Khi sử dụng con trỏ tới mảng  
C. Khi con trỏ trỏ tới biến toàn cục  
D. Khi sử dụng con trỏ NULL  
**Đáp án:** A

**Câu 36:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int arr[] = {1, 2, 3};

int \*p1 = arr, \*p2 = arr + 2;

printf("%ld", p2 - p1);

A. 2  
B. 1  
C. 0  
D. Lỗi biên dịch  
**Đáp án:** A

**Câu 37:** Con trỏ hàm dùng để:  
A. Truyền tham số kiểu hàm vào hàm khác  
B. Lưu địa chỉ của một hàm  
C. Gọi hàm thông qua con trỏ  
D. Tất cả các ý trên  
**Đáp án:** D

**Câu 38:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int x = 100;

int \*p = &x;

\*p = \*p + 1;

printf("%d", x);

A. 100  
B. 101  
C. Địa chỉ của x  
D. Lỗi runtime  
**Đáp án:** B

**Câu 39:** Đoạn mã sau có kết quả gì?

|  |
| --- |
|  |

void func(int \*\*p) {

\*\*p = 20;

}

int main() {

int a = 10;

int \*p = &a;

func(&p);

printf("%d", a);

return 0;

}

A. 10  
B. 20  
C. Địa chỉ của a  
D. Lỗi runtime  
**Đáp án:** B

**Câu 40:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int x = 10;

int \*p1 = &x, \*p2 = p1;

\*p2 = 20;

printf("%d", x);

A. 10  
B. 20  
C. Địa chỉ của x  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 41:** Đoạn mã sau in ra gì?

|  |
| --- |
|  |

int a = 5, b = 15;

int \*p = &a;

\*p = b;

printf("%d %d", a, b);

A. 5 15  
B. 15 15  
C. 15 5  
D. Lỗi runtime  
**Đáp án:** B

**Câu 42:** Khi truyền một con trỏ vào hàm, giá trị mà con trỏ trỏ tới có thể:  
A. Thay đổi bên trong hàm  
B. Không thay đổi  
C. Phụ thuộc vào giá trị của tham số  
D. Không xác định  
**Đáp án:** A

**Câu 43:** Điều gì xảy ra khi không giải phóng bộ nhớ cấp phát bằng malloc?  
A. Bộ nhớ bị rò rỉ (memory leak)  
B. Lỗi biên dịch  
C. Lỗi runtime  
D. Bộ nhớ tự động giải phóng  
**Đáp án:** A

**Câu 44:** Con trỏ void \* cần phải làm gì trước khi sử dụng?  
A. Ép kiểu về kiểu dữ liệu cụ thể  
B. Sử dụng trực tiếp  
C. Không thể sử dụng  
D. Chỉ sử dụng trong malloc  
**Đáp án:** A

**Câu 45:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int arr[] = {1, 2, 3, 4};

int \*p = arr + 1;

printf("%d", \*(p - 1));

A. 1  
B. 2  
C. 0  
D. Lỗi runtime  
**Đáp án:** A

**Câu 46:** Đoạn mã sau có lỗi gì?

|  |
| --- |
|  |

int \*p = NULL;

\*p = 5;

A. Lỗi biên dịch  
B. Không lỗi  
C. Lỗi runtime do truy cập con trỏ NULL  
D. Không xác định  
**Đáp án:** C

**Câu 47:** Đoạn mã sau in ra gì?

|  |
| --- |
|  |

char \*str = "Hello";

printf("%c", \*(str + 4));

A. H  
B. o  
C. e  
D. Lỗi runtime  
**Đáp án:** B

**Câu 48:** Đoạn mã sau có lỗi gì?

|  |
| --- |
|  |

int arr[5] = {1, 2, 3, 4, 5};

int \*p = arr + 5;

printf("%d", \*p);

A. Lỗi biên dịch  
B. Không lỗi, nhưng hành vi không xác định  
C. Lỗi runtime do truy cập ngoài mảng  
D. Không lỗi, in ra 0  
**Đáp án:** B

**Câu 49:** Con trỏ hàm được khai báo như thế nào?  
A. int (\*fptr)(int, int);  
B. int \*fptr(int, int);  
C. int fptr(\*int, \*int);  
D. void \*fptr();  
**Đáp án:** A

**Câu 50:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
|  |

int x = 10;

int \*p = &x;

int \*\*pp = &p;

\*\*pp = 15;

printf("%d", x);

A. 10  
B. 15  
C. Địa chỉ của x  
D. Lỗi runtime  
**Đáp án:** B

|  |
| --- |
|  |